What is claimed is:

1. A method of pausing execution of instructions in a thread, comprising:

determining if a next instruction for a first thread is an instruction of a first type; and

preventing instructions of said first thread from being processed for execution for a period of time while instructions from a second thread can be processed for execution.

2. The method of claim 1 wherein said next instruction of a first type is a SET instruction and a following instruction is a READ instruction

3. The method of claim 2 wherein said SET instruction causes a value to be stored in memory for said first thread.

4. The method of claim 3 further comprising:

processing said READ instruction for execution when said value stored in memory is reset.

5. The method of claim 4 wherein said value stored in memory is reset when said SET instruction is retired.

6. A method of pausing execution of instructions in a thread, comprising:

determining if a next instruction for a first thread is an instruction of a first type;

initiating a counter; and

preventing instructions of said first thread from being processed for execution until said counter reaches a predetermined value.

7. The method of claim 6 wherein said instruction includes an operand and said initiating step includes loading said counter with said operand.

8. The method of claim 6 wherein instructions for a second thread can be processed for execution while instructions of said first thread are prevented from being processed.

9. A method of reducing power consumption in a processor system comprising:

receiving a pause instruction from a first thread at a decode unit in said processor system;

preventing instructions of said first thread from being processed for execution for a period of time while instructions from a second thread can be processed for execution.

10. The method of claim 9 wherein execution of software code causes said pause instruction to be received at said decode unit.

11. An apparatus pausing execution of instructions in a thread, comprising:

a decode unit to determine if a next instruction for a first thread is an instruction of a first type, said decode unit to prevent instructions of said first thread from being processed for execution for a period of time while instructions from a second thread can be processed for execution.

12. The apparatus of claim 11 wherein said next instruction of a first type is a SET instruction and a following instruction is a READ instruction

13. The apparatus of claim 12 further comprising:

a memory, wherein said SET instruction causes a value to be stored in memory for said first thread.

14. The apparatus of claim 13 wherein said decode unit processes said READ instruction for execution when said value stored in memory is reset.

15. The apparatus of claim 14 further comprising:

a retire unit coupled to said decode unit wherein said retire unit causes said value stored in memory to be reset when said SET instruction is retired by said retire unit.

16. An apparatus for pausing execution of instructions in a thread, comprising:

a decode unit to determine if a next instruction for a first thread is an instruction of a first type;

a counter that is initiated when said next instruction for a first thread is an instruction of a first type, said decode unit to prevent instructions of said first thread from being processed for execution until said counter reaches a predetermined value.

17. The apparatus of claim 16 wherein said instruction includes an operand to be loaded into

said counter.

18. The apparatus of claim 16 wherein instructions for a second thread can be processed for execution while instructions of said first thread are prevented from being processed.

19. An apparatus for reducing power consumption in a processor system comprising:

a processor system including

a decode unit to receive a pause instruction from a first thread in said processor system, said decode unit to prevent instructions of said first thread from being processed for execution for a period of time while instructions from a second thread can be processed for execution.

20. The apparatus of claim 19 wherein execution of software code at said processor system causes said pause instruction to be received at said decode unit.